

Problem 5 - Coin Combinations

John is counting the loose change found in his pocket, and wonders to himself about the various combinations of coin values that can make up a specific monetary value. You have **infinite** pennies, nickels, dimes, and quarters. To help John out, write a program that will output all combinations of coins such that the total is equal to the input cent value.

For example: 3 quarters, 2 nickels, and 4 pennies is one of many ways to make up 99 cents.

Key Points

- The output should begin with a combination of coin values that make up the input cent value, and each new coin combination should begin on a new line.
- Each coin value must be separated by a comma with **no** spaces.
- Each combination output line must end with a coin value (no ending comma).
- The output file should also include the total number of combinations found; for example, "There are # combinations." where # is the number of combinations. See example output for more information.
- The input monetary value must be in **cents**.

Input

The input will consist of one input line delimited by a new line character. The value must be in a cent value. For example, to input \$1.50 you would enter 150.

Output

The output will consist of a combination line that is delimited by a new line character. Each coin should be separated by a comma with no spaces. Each combination should be ordered from the lowest coin denomination first to the highest coin denomination last. And the lines should be ordered from longest combination first to shortest combination last. The total number of combinations should be displayed after the combinations are outputted, and spaced with a new line.

Sample Input

```
10
```

Example output

```
1,1,1,1,1,1,1,1,1,1
1,1,1,1,1,5
5,5
10
```

There are 4 combinations.